



Transforme seu Código



Conheça as funções para deixar o seu código mais
organizado e legível

FUNÇÕES EM PYTHON

Guia Prático de Funções em Python

As funções são blocos de código reutilizáveis que ajudam a tornar seu código mais organizado e legível. Vamos explorar as principais regras para criar e utilizar funções em Python com exemplos práticos.



01

DEFININDO UMA FUNÇÃO

Para definir uma função em Python, usamos a palavra-chave `def`.

COMO DEFINIR UMA FUNÇÃO



Para definir uma função em Python, usamos a palavra-chave `def` seguida do nome da função e parênteses.

```
● ● ● Como Definir uma Função  
  
def saudacao():  
    print("Olá, Mundo!")
```

Imagine que você deseja criar uma função para saudar um usuário específico. Exemplo:

```
● ● ● Função de saudação  
  
def saudacao_usuario(nome):  
    print(f"Olá, {nome}!")  
  
saudacao_usuario("Alice")
```

02

PARÂMETROS E ARGUMENTOS

As funções podem receber informações através de parâmetros.

FUNÇÕES COM PARÂMETROS



As funções podem receber informações através de parâmetros, permitindo que você passe valores específicos.

```
Funções com Parâmetros

def soma(a, b):
    return a + b
```

Criando uma função para calcular o total de uma compra.
Exemplo:

```
Total de uma compra

def calcular_total(preco, quantidade):
    return preco * quantidade

total = calcular_total(10.99, 3)
print(f"Total da compra: R${total}")
```

03

Valores Padrão Parâmetros

Você pode definir valores padrão para parâmetros

PARÂMETROS COM VALORES PADRÃO



Você pode definir valores padrão para parâmetros, tornando-os opcionais.

```
Definindo valores padrão

def saudacao(nome="Visitante"):
    print(f"Olá, {nome}!")
```

Um sistema de cadastro onde nem todos os campos são obrigatórios. Exemplo:

```
Sistema com campos obrigatórios

def cadastrar_usuario(nome, email="Não informado"):
    print(f"Usuário: {nome}, Email: {email}")

cadastrar_usuario("Alice")
cadastrar_usuario("Bob", "bob@example.com")
```


04

RETORNANDO VALORES

Funções podem retornar valores usando a palavra-chave `return`.

FUNÇÕES QUE RETORNAM VALORES



Funções podem retornar valores usando a palavra-chave `return`.

```
Return

def quadrado(numero):
    return numero ** 2
```

Calculando o quadrado de um número fornecido pelo usuário. Exemplo:

```
Return número fornecido pelo usuário

def calcular_quadrado(numero):
    return numero ** 2

resultado = calcular_quadrado(5)
print(f"O quadrado de 5 é {resultado}")
```

05

FUNÇÕES ANINHADAS

Você pode definir funções dentro de outras funções

DEFININDO FUNÇÕES DENTRO DE OUTRAS



Você pode definir funções dentro de outras funções para organizar melhor seu código.



Funções Aninhadas

```
def saudacao_personalizada(nome):  
    def mensagem():  
        return "Bem-vindo"  
    return f"{mensagem()}, {nome}!"
```

Uma função que cria uma mensagem personalizada de boas-vindas. Exemplo:



Mensagem personalizada

```
def criar_mensagem_boas_vindas(nome):  
    def mensagem():  
        return "Olá"  
    return f"{mensagem()}, {nome}! Estamos  
felizes em tê-lo aqui."  
  
mensagem_boas_vindas =  
criar_mensagem_boas_vindas("Carlos")  
print(mensagem_boas_vindas)
```

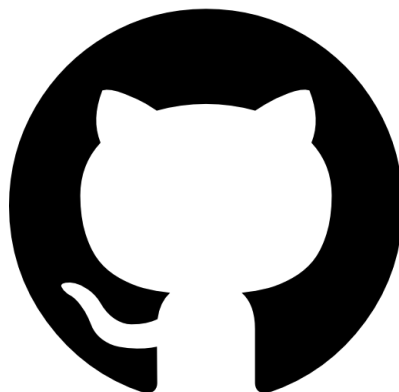
Agradecimientos

OBRIGADA POR LER ATÉ AQUI



Esse E-book foi gerado por AI, e diagramado por humano.

Esse conteúdo foi gerado com fim colaborativo para o projeto Criando um E-book com ChatGPT & MidJourney proposto pelo instrutor Felipe Aguiar Tech Educator, DIO.



O passo a passo se encontra no Github:

<https://github.com/Cristina-santos89/Criando-um-Ebook-com-ChatGPT-MidJourney>
